Euler theorem:

f=input('Enter your function: ') ;

x0=input('Enter initial value of independent variable: ') ;

y0=input('Enter initial value of dependent variable: ') ;

h=input('Enter step size: ') ;

xn=input('Enter the point in which you want to evaluate: ') ;

n=(xn-x0)/h ;

x(1)=x0 ;

y(1)=y0 ;

for i=1:n

y(i+1)=y(i)+h\*f(x(i),y(i)) ;

x(i+1)=x0+i\*h ;

fprintf("y(%.2f) = %.4f\n",x(i+1),y(i+1));

end

Runge-Kutta:

f=input('Enter your function: ') ;

x0=input('Enter initial value of independent variable: ') ;

y0=input('Enter initial value of dependent variable: ') ;

h=input('Enter step size: ') ;

xn=input('Enter the point in which you want to evaluate: ') ;

n=(xn-x0)/h ;

x(1)=x0 ;

y(1)=y0 ;

for i=1:n

x(i+1)=x0+i\*h ;

k1=h\*f(x(i),y(i)) ;

k2=h\*f(x(i+1),y(i)+k1) ;

y(i+1)=y(i)+.5\*(k1+k2) ;

fprintf("y(%.2f) = %.4f\n",x(i+1),y(i+1));

end

Gauss elimination:

A=input("Enter the co-efficient matrix: ") ;

B=input('Enter source vector: ') ;

N=length(B) ;

aug=[A B] ;

for j=1:N-1

for i=j+1:N

m=aug(i,j)/aug(j,j) ;

aug(i,:)=aug(i,:)-m\*aug(j,:) ;

end

end

aug

X(N)=aug(N,N+1)/aug(N,N) ;

for i=N-1:-1:1

X(i)=(aug(i,N+1)-aug(i,i+1:N)\*X(i+1:N))/aug(i,i) ;

end

X

Gauss Jacobi:

A=input('Enter thr co-efficient matrix: ')

B=input('Enter Source Vector B: ')

P=input('Enter initial gauss vecotor: ')

n=input('Enter number of iterations: ')

N=length(B)

X=zeros(N,1)

for j=1:n

for i=1:N

X(i)=(B(i)/A(i,i))-(A(i,[1:i-1,i+1:N])\*P([1:i-1,i+1:N]))/A(i,i);

end

fprintf('Iteration no. %d\n,j',j)

X

if abs(X-P)<e

break ;

end

P=X ;

end

Gauss Seidal

A=input('Enter thr co-efficient matrix: ') ;

B=input('Enter Source Vector B: ') ;

P=input('Enter initial gauss vecotor: ') ;

n=input('Enter number of iterations: ') ;

e=input('Enter your tolerance: ') ;

N=length(B) ;

X=zeros(N,1) ;

Y=zeros(N,1) ;

for j=1:n

for i=1:N

X(i)=(B(i)/A(i,i))-(A(i,[1:i-1,i+1:N])\*P([1:i-1,i+1:N]))/A(i,i);

end

X

if(abs(Y-X)<e)

break;

end

Y=X ;

P=X ;

end

Trapezoidal:

f=input('Enter the equation: ');

x0=input('Enter the lower limit: ')

xn=input('Enter the number of intervals: ')

h=(xn-x0)/n ;

sum=f(x0)+f(xn) ;

for(i=1:n-1)

sum=sum+2\*f(x0+i\*h) ;

end

I=(h\*sum)/2 ;

Cramer’s rule

D = input('Enter your coefficient matrix: ') ;

b = input('Enter source vector: ') ;

N = length(b) ;

X=zeros(N,1) ;

d=det(A);

Aold=A;

if d~=0

for i=1:N

A(:,i)=b;

X(i)=det(A)/d;

A=Aold;

end

disp("Solution using cramer's rule is ")

X

else

disp('Cramer rule not applicable')

end

Gauss elimination:

A=input("Enter the co-efficient matrix: ") ;

B=input('Enter source vector: ') ;

N=length(B) ;

aug=[A B] ;

for j=1:N-1

for i=j+1:N

m=aug(i,j)/aug(j,j) ;

aug(i,:)=aug(i,:)-m\*aug(j,:) ;

end

end

aug

X(N)=aug(N,N+1)/aug(N,N) ;

for i=N-1:-1:1

X(i)=(aug(i,N+1)-aug(i,i+1:N)\*X(i+1:N))/aug(i,i) ;

end

X